Auto Process NY Covid-19 Case Data

John Minter

Started 2020-03-27, Last modified: 2020-04-07

Contents

Plot Total COVID-19 Cases

Set up the analysis. Use the `bVerbose` variable as a flag to decide whether or not to print diagnostic data. The default is `FALSE`.

```
bVerbose <- TRUE

if(bVerbose){
   print(getwd())
}
```

```
[1] "/Users/jrminter/Documents/git/anaCovid19/R"
```

Pull down the data

```
temp <- tempfile()
download.file("https://github.com/nytimes/covid-19-data/archive/master.zip",temp)
us_counties <- read_csv(unz(temp, "covid-19-data-master/us-counties.csv"))
```

```
Parsed with column specification:
cols(
  date = col_date(format = ""),
  county = col_character(),
  state = col_character(),
  fips = col_character(),
  cases = col_double(),
  deaths = col_double()
)
```

```
path <- paste0(here(), "/us-counties.csv")
write_csv(us_counties, path, append = FALSE, col_names=TRUE)
us_states <- read_csv(unz(temp, "covid-19-data-master/us-states.csv"))
```

```
Parsed with column specification:
cols(
  date = col_date(format = ""),
  state = col_character(),
  fips = col_character(),
  cases = col_double(),
  deaths = col_double()
)
```

```r
path <- paste0(here(), "/us-states.csv")
write_csv(us_states, path, append = FALSE, col_names=TRUE)
unlink(temp)
tail(us_counties)
```

```
# A tibble: 6 x 6
  date       county     state   fips  cases deaths
  <date>     <chr>      <chr>   <chr> <dbl>  <dbl>
1 2020-04-07 Sheridan   Wyoming 56033    12      0
2 2020-04-07 Sublette   Wyoming 56035     1      0
3 2020-04-07 Sweetwater Wyoming 56037     6      0
4 2020-04-07 Teton      Wyoming 56039    44      0
5 2020-04-07 Uinta      Wyoming 56041     3      0
6 2020-04-07 Washakie   Wyoming 56043     4      0
```

```r
tail(us_states)
```

```
# A tibble: 6 x 5
  date       state          fips  cases deaths
  <date>     <chr>          <chr> <dbl>  <dbl>
1 2020-04-07 Virgin Islands 78       45      1
2 2020-04-07 Virginia       51     3333     69
3 2020-04-07 Washington     53     8682    409
4 2020-04-07 West Virginia  54      412      4
5 2020-04-07 Wisconsin      55     2578     94
6 2020-04-07 Wyoming        56      221      0
```

Next, load the data file and extract what we need using functions from the `dplyr` package to create a `tibble` of values (an enhanced R dataframe that works nicely with the `tidyverse` collection of R packages by Hadley Wickham.) We will use the `kable` function from the `knitr` package to get a nice looking table. We really only want the last few values…

```r
pa_states <- paste0(here(), "/us-states.csv")
df <- read.csv(pa_states, header = TRUE, sep = ",")
df$date <- format(as.Date(df$date), "%m-%d")
tib <- as_tibble(df)
tib <- tib %>% filter(state == "New York")
tib  %>% select(date, cases, deaths) -> new_york

if(bVerbose){
   print(length(new_york$date))
   print(typeof(new_york$date[1]))
}
```

```
[1] 38
[1] "character"
```

```r
new_york$date <- as.Date(tib$date, "%m-%d")

if(bVerbose){
   head(new_york)
}
```

```
# A tibble: 6 x 3
  date       cases deaths
  <date>     <int>  <int>
1 2020-03-01     1      0
2 2020-03-02     1      0
3 2020-03-03     2      0
4 2020-03-04    11      0
5 2020-03-05    22      0
6 2020-03-06    44      0
```

```r
kable(tail(new_york))
```

date	cases	deaths
2020-04-02	92770	2653
2020-04-03	102870	2935
2020-04-04	114996	3568
2020-04-05	122911	4161
2020-04-06	130703	4758
2020-04-07	140081	5563

```r
if(bVerbose){
   print(typeof(tib$date))
   print(tib$date)
}
```

```
[1] "character"
 [1] "03-01" "03-02" "03-03" "03-04" "03-05" "03-06" "03-07" "03-08" "03-09"
[10] "03-10" "03-11" "03-12" "03-13" "03-14" "03-15" "03-16" "03-17" "03-18"
[19] "03-19" "03-20" "03-21" "03-22" "03-23" "03-24" "03-25" "03-26" "03-27"
[28] "03-28" "03-29" "03-30" "03-31" "04-01" "04-02" "04-03" "04-04" "04-05"
[37] "04-06" "04-07"
```

Next, we compute the death rate. We really only need the last value in the table.

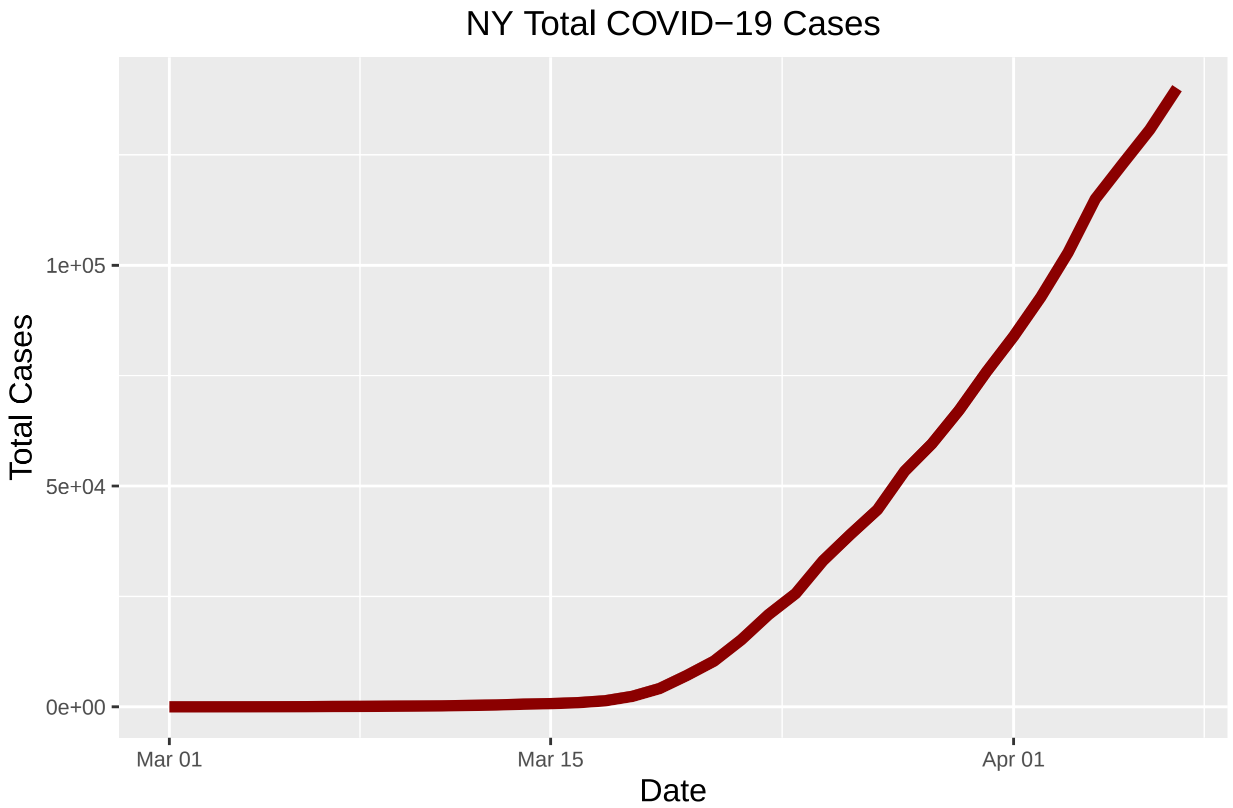
```r
tot_deaths <- max(tib$deaths)
tot_cases <- max(tib$cases)

death_rate_pct <- 100*tot_deaths/tot_cases

death_rate_pct <- round(death_rate_pct, digits = 2)
```

New York's COVID-19 is 3.97 percent.

Next, we plot the curve.

```r
plt_ny_tot <- ggplot(new_york, aes(x=date, y=cases)) +
              geom_line(colour='darkred', size=2) +
              xlab("Date") +
              ylab("Total Cases") +
              ggtitle("NY Total COVID-19 Cases") +
              labs(caption = 'Data from Johns Hopkins') +
              scale_x_date() +
              # theme_minimal() +
              theme(axis.text=element_text(size=8),
              axis.title=element_text(size=12),
              plot.title=element_text(hjust = 0.5)) +
              NULL
```

```r
print(plt_ny_tot)
```



Data from Johns Hopkins

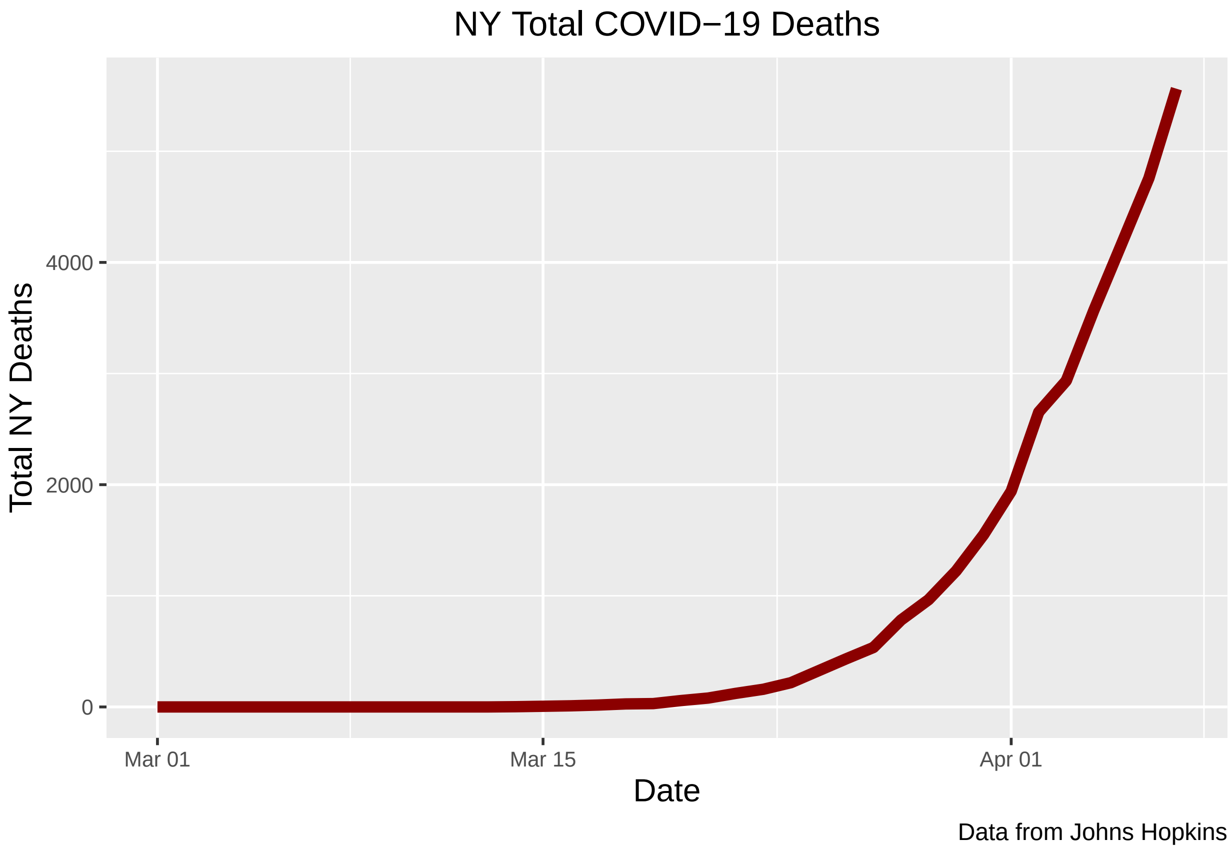
Sadly, the curve has not flattened yet. Recall that this is for all of New York State and is **dominated by cases in New York City**.

We will save the plot as in both the png and jpg formats.

```r
strOutPng <- "plt/Covid-19-Total-Cases-NY.png"

# we want the png to be close to 1024x768...
ggsave(plt_ny_tot, file=strOutPng, width=9.0, height=6.0,
       units="in", dpi=113.7778)

strOutJpg <- "plt/Covid-19-Total-Cases-NY.jpg"

# we want the png to be close to 1024x768...
ggsave(plt_ny_tot, file=strOutJpg, width=9.0, height=6.0,
       units="in", dpi=113.7778)
```

Plot NY COVID Deaths

```r
plt_ny_deaths <- ggplot(new_york, aes(x=date, y=deaths)) +
                 geom_line(colour='darkred', size=2) +
                 xlab("Date") +
                 ylab("Total NY Deaths") +
                 ggtitle("NY Total COVID-19 Deaths") +
                 labs(caption = 'Data from Johns Hopkins') +
                 scale_x_date() +
                 # theme_minimal() +
                 theme(axis.text=element_text(size=8),
                 axis.title=element_text(size=12),
                 plot.title=element_text(hjust = 0.5)) +
                 NULL

print(plt_ny_deaths)
```



We will save the plot as in both the `png` and `jpg` formats.

```
strOutPng <- "plt/Covid-19-Total-Deaths-NY.png"

# we want the png to be close to 1024x768...
ggsave(plt_ny_deaths, file=strOutPng, width=9.0, height=6.0,
       units="in", dpi=113.7778)

strOutJpg <- "plt/Covid-19-Total-Deaths-NY.jpg"

# we want the png to be close to 1024x768...
ggsave(plt_ny_deaths, file=strOutJpg, width=9.0, height=6.0,
       units="in", dpi=113.7778)
```

Plot COVID New York New Cases

Make a tibble of new cases

```
new_cases <- diff(tib$cases)
the_date <- as.Date(tib$date[-1], "%m-%d")
print(class(the_date))
```

```
[1] "Date"
```

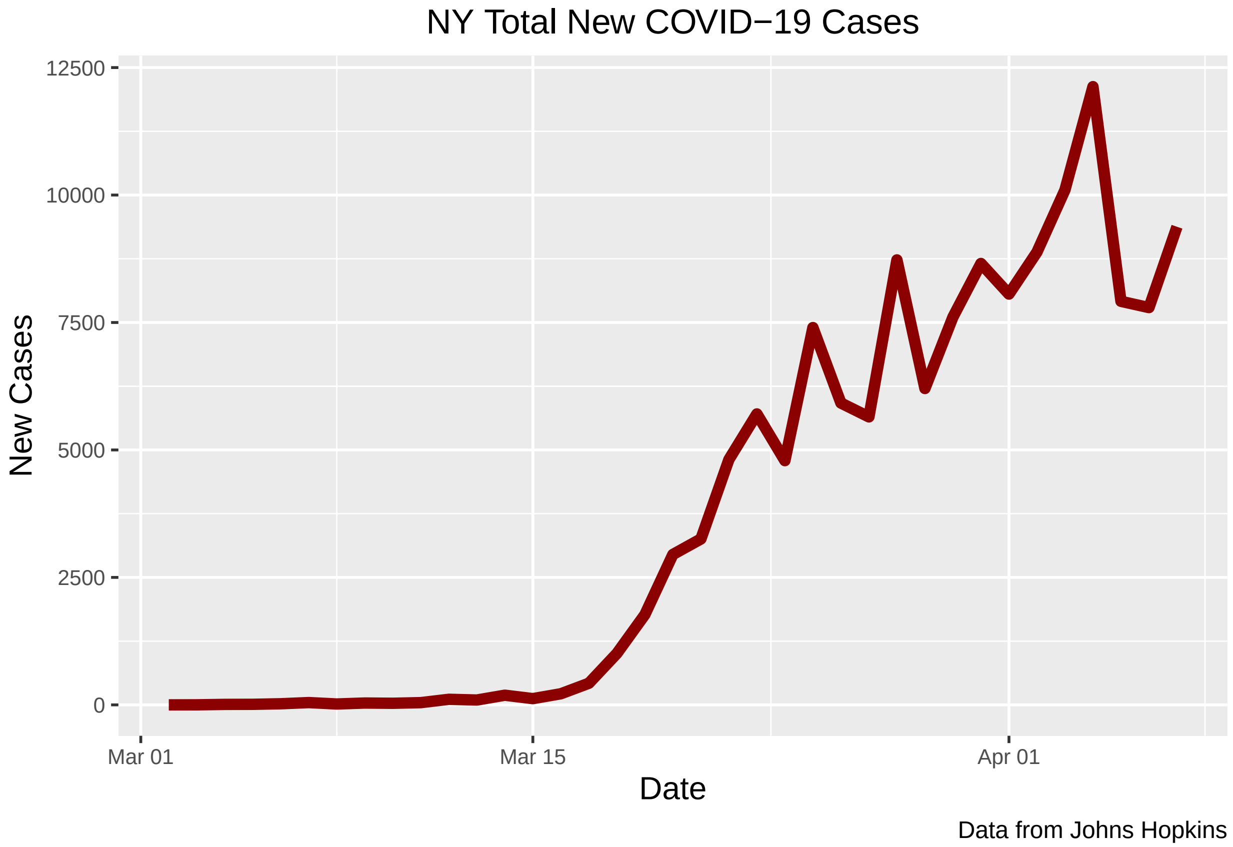
```
new_cases_tbl <- data.frame(the_date, new_cases)
print(head(new_cases_tbl))
```

```
    the_date new_cases
1 2020-03-02         0
2 2020-03-03         1
3 2020-03-04         9
4 2020-03-05        11
5 2020-03-06        22
6 2020-03-07        45
```

```
print(tail(new_cases_tbl))
```

```
     the_date new_cases
32 2020-04-02      8881
33 2020-04-03     10100
34 2020-04-04     12126
35 2020-04-05      7915
36 2020-04-06      7792
37 2020-04-07      9378
```

```
plt_ny_new_cases <- ggplot(new_cases_tbl, aes(x=the_date, y=new_cases)) +
                    geom_line(colour='darkred', size=2) +
                    xlab("Date") +
                    ylab("New Cases") +
                    ggtitle("NY Total New COVID-19 Cases") +
                    labs(caption = 'Data from Johns Hopkins') +
                    scale_x_date() +
                    theme(axis.text=element_text(size=8),
                    axis.title=element_text(size=12),
                    plot.title=element_text(hjust = 0.5)) +
                    NULL

print(plt_ny_new_cases)
```



Save the New Cases Plot

```
strOutPng <- "plt/Covid-19-New-Cases-NY.png"

# we want the png to be close to 1024x768...
ggsave(plt_ny_new_cases, file=strOutPng, width=9.0, height=6.0,
       units="in", dpi=113.7778)

strOutJpg <- "plt/Covid-19-New-Cases-NY.jpg"

# we want the png to be close to 1024x768...
ggsave(plt_ny_new_cases, file=strOutJpg, width=9.0, height=6.0,
       units="in", dpi=113.7778)
```